Team Members: Alex Buchanan, Jess Lai, Kevin Holmes, Mike Danino

Project Leader: Kevin Holmes

Project Name: Game of Death

Description:

Conway’s Game of Life is a well known algorithm in the CS field for being among the first simulations. The purpose was to be a proof of concept of simulating cellular automata.

Alex and Kevin had worked together during Mobile App 1 to implement this algorithm for viewing on Android. Finishing ahead of time, further features were added including:

* Saving the entire grid to SQL
* Saving interesting sections of a grid separately to SQL
* The ability to paste saved sections onto the screen, wherever they like by dragging around the object
* Variable simulation speed control
* Variable grid sizes (actively redrawing as the user modified the value)
* Directly drawing the cells onto the grid in a sort of “finger-painting” mode.

The completed product was visually appealing and interesting but lacked the user interactivity that we were aiming for. Not much could be done, as it was just a simple simulation good for watching, not necessarily playing with.

Game of Death is our new idea to expand on and accomplish what we wanted if we were given another few months with the original project.

To achieve the level of user interactivity we want, we are converting the entire idea (and writing from scratch) into a turn-based strategy game. This game style is nothing new; chess, checkers, go, and many others have been analyzed and mastered over the years. This game however will require the user to think dynamically; How will the seeds placed on the board interact with one another? With other strategy games, your pieces are your pieces, and the enemy has their own. Using the algorithm at the core of Game of Life, Game of Death allows for a further level of complexity. The user can defeat themselves if they do not anticipate how their actions will affect their own pieces. One false move and they could cause a collapse of their entire colony.

For this reason, we have decided it would prove challenging enough, while the backing logic understood well enough, that it would be a worthwhile game to create.

**Pro’s:**

* 2 of us on the team have previous experience with Conway’s Game of Life from Mobile App 1.
* Fun and interesting game that would provide a good challenge to both implement and play.
* Really haven’t seen the simulation turned into a game before.

**Con’s:**

* Taking the already existing algorithm and creating a game that is both easy enough for a player understand while still staying faithful enough to the original algorithm while require a lot of testing and adjustments
* The logic involved in manipulating the algorithm to have opposing sides is a complex topic and difficult task.

**Game Logic:**

* Player is given currency at the beginning of the game, which replenishes by a constant, small amount on every turn. Currency is used to buy “seeds” (shown in next slide)
* Enemy is not AI – All enemy seeds are placed on the board at the beginning of a round
* Win condition – After a given number of rounds, the side with the most cells alive wins
* Features
  + Opposing cells annihilate one another
  + Terrain – background art will be added, with certain grid locations being blocked off and inaccessible
  + Multiple levels which ramp up difficulty
  + Animated seed selection allows the player to see how any given seed evolves over the course of multiple turns
  + Challenge is for the player to anticipate how those cells will come together against the enemy; or possibly do themselves harm.

**Technical Aspects:**

* The stepper algorithm in charge of the individual cells won’t have to iterate as much as the previous app.
  + Previous app allocated a grid size with a 1:4 pixels to cell ratio. A modern (mid-range) phone’s resolution is 1920x1080. That’s a lot of iterations. The look and feel of this version allows for much larger cells, and less performance issues.
* The main game board implements the SurfaceView class
  + This enables OpenGL acceleration, allowing for further refinements to the overall look and feel of the game; shaders, textures, etc.
  + Drawing thread is separate from the logic thread, which allows the application to proceed without ever blocking itself.
  + Touch input handlers are applied to the main thread, which lets the drawing thread give user the previews of where the the seeds will be dropped. This preview paste object will snap into the grid locations as the user moves their finger about the screen.

**Time Tables:**

**Alpha:**

* Expected near end of February/beginning of March
* 1 initial (easy mode) level
* Core gameplay implemented
  + At least 3 seeds available
  + Opposing cells annihilate each other
  + Pieces can be dragged and pasted by the player
  + Currency implemented
  + “Finish Turn” button implemented and allows normal progression of game

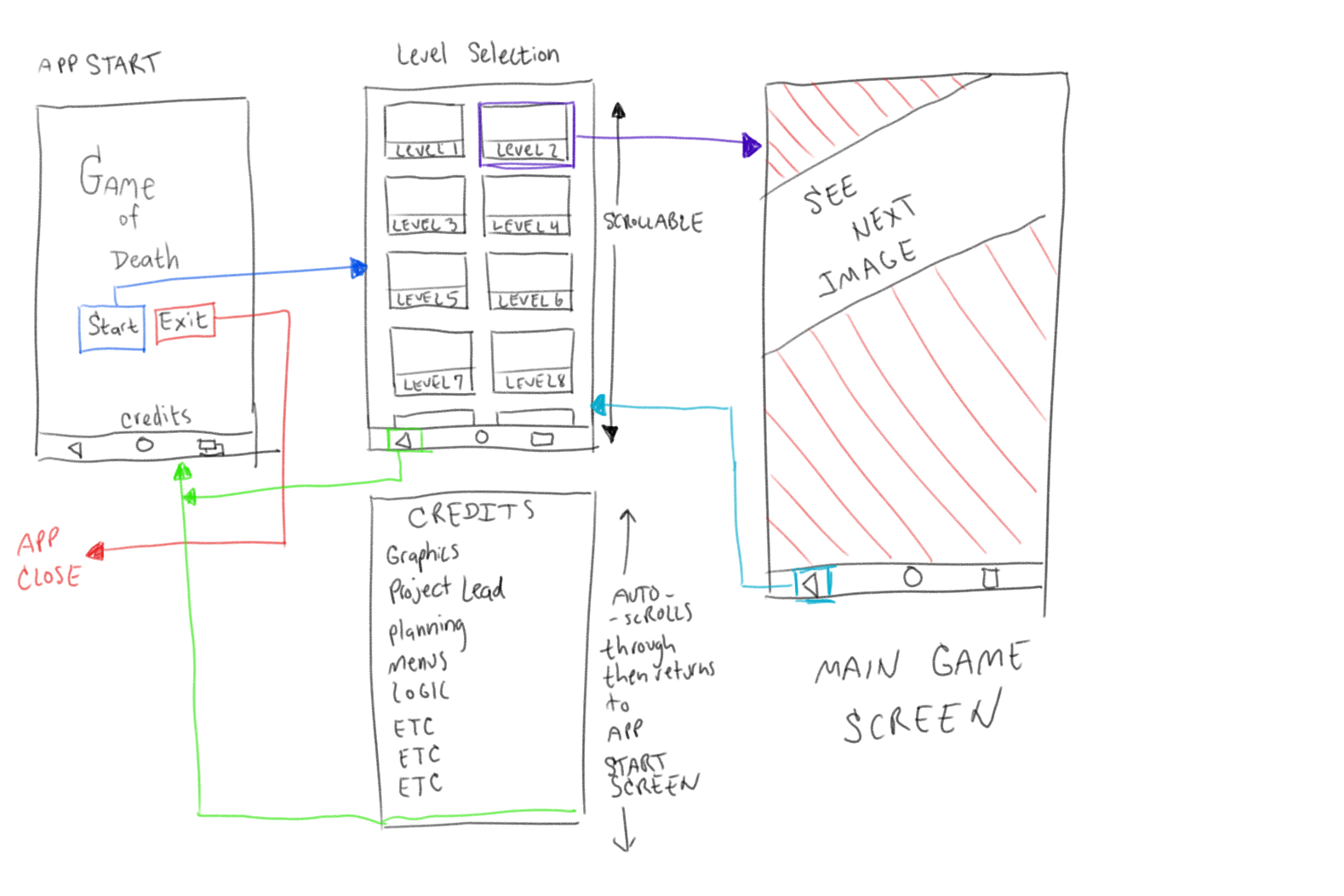
**Beta:**

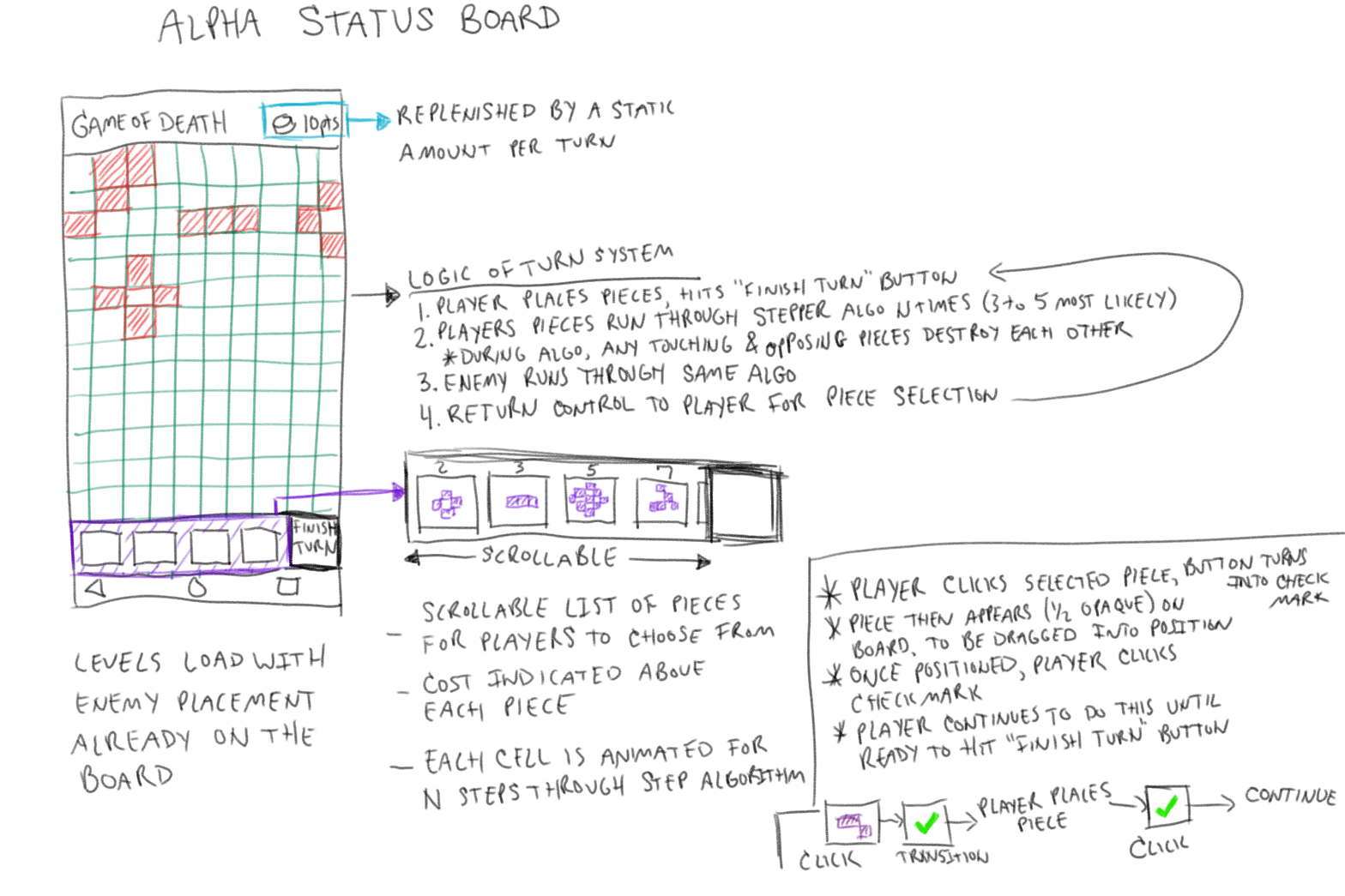
* Expected near end of March/beginning of April
* More than 1 level available, with increasing difficulty
* Gameplay features refined
  + At least 10 seeds available
  + Currency system stabilized to what seems fair but remaining challenging
  + Multiple terrains available
  + Seed buttons animated
  + Initial artwork applied to menus, cells, and terrain

**Final:**

* Expected at end of semester
* All levels available, goal is ~6-8
* Gameplay features completed
  + All expected and strategy-worthwhile seeds implemented
  + Scoring system refined and highscores implemented
  + Terrains refined to final forms
  + Artwork for terrain, menus, and cells completed

**Mockups:**

****

****